Commonwealth of Massachusetts
Health and Human Services

Mass.Gov

Methodology User Guide

Version 1.0

Revision History

Date	Version	Description	Author
Aug 15, 2003	1.0	Initial Version	Rita El-Khoury
Feb 3, 2005	1.1	Consistency Review	Pasquale Iocola

Table of Contents

Table Of Figures

Table Of Tables

Methodology User Guide

1. Introduction

1.1 Purpose

This document is the primary detailed description of the Executive Office of Health and Human Services (EOHHS) development Methodology that was created as part of the Virtual Gateway initiative (hereafter VG). The EOHHS Methodology is prescribed and maintained by the Project Management Office (PMO). It provides a set of guidelines that outline the workings of a team-oriented approach to specifying, constructing, and assembling software and hardware components into a working system that meets well-defined requirements.

As per the PMO charter, this user guide should be read, understood and followed to the extent possible by all project team members involved in the development of a system that is in line with the VG initiative. Project team members need to become familiar with the guidelines, principles, and procedures contained within this user guide. In addition, the concepts described in this document may be tailored to accommodate project size, complexity, and risk.

The guidelines presented in this user guide are not static, and evolve as business, technology, and information systems need change. It is equally important for the users of this user guide to provide feedback, suggestions, and ideas on how to improve and enhance this material and its content. Suggested enhancements can be relayed to the PMO for consideration.

1.2 How This Document is Organized

This user guide starts off by highlighting the key Methodology concepts. Then it goes on to describe each of the project phases in 4 main sections. Each main section provides the details of the activities and deliverables required for the phase. For each phase, the following sub-sections are included:

- *Definition* - Provides a definition of the phase

- *Goal* - Identifies the advantages/disadvantages and the risks/benefits of performing the activities of the phase.

- *Exit Criteria* - Provides a summary table as well as detailed sub-sections outlining what needs to be done by each team member to successfully complete the phase and move on to the next phase.

At the end, a Glossary section is included to define the terms and acronyms used in this user guide. In addition a Tools and Job Aids section is included to provide links to Methodology support documentation used to assist the project team members in the completion of their assigned tasks such as templates.

2. Methodology Key Concepts

This EOHHS Methodology user guide describes the application development practices at EOHHS. It is based on *The Unified Software Development Process,* I. Jacobsen, G. Booch, J. Rumbaugh, and uses UML (*The Unified Modeling Language User Guide*, I Jacobsen, G. Booch, J. Rumbaugh) as the standard modeling language.

The Methodology is based on 6 main themes:

1. *Use-case Driven* - Use cases provide a means for capturing functional requirements, organizing activities, and keeping the entire team focused on the end result.

2. *Architecture-Centric* - The central technical activity is architecture, which is developed and validated early, and the rest of the system is built around it.

3. *Emphasis on Systemic Qualities* – The system is architected to meet the Service Level Requirements at each layer and each tier. The Systemic Qualities are prioritized, architected for and validated early.

4. *Iterative and Incremental* - The bigger system is evolved from a series of smaller systems, each of which extends the previous one.

5. *Focus on Early Risk Mitigation* - Risks are identified early and tracked. Early activities are focused on risk reduction.

6. *Focus on High vs. Low Value Artifacts* – Intermediate and transient artifacts such as analysis models and design documents are not kept up-to-date and formalized. More important artifacts such as the System Architecture Document (SAD) are signed off and maintained.

The Methodology builds on the concepts of phases and workflows. A phase is all about focus and priority. The partitioning of the project timeline into phases serves to clarify and emphasize the priorities both internally and externally to the project. Each phase is defined by its deliverables, which drive the activities that must occur within that phase. A workflow is a grouping of closely related activities. See the sections below for details on the Methodology phases and workflows.

The moves between phases are considered major milestones. The end of each phase is accompanied by a go/no-go decision. Each phase consists of one or more iterations. The moves between iterations are considered minor milestones that encompass intermediate internal or external releases.

2.1 Phases

Four phases are defined for each product release. The four phases are:

1. *Inception* is the first phase. It defines the project's scope, its risks, and estimates its major milestones. It is also the beginning of requirements gathering and defining the architecture strategy.

2. *Elaboration* follows Inception. It focuses on mitigating risks through architectural prototyping and on fleshing out the requirements.

3. *Construction* follows Elaboration. This is where the bulk of the system's functionality is designed and built.

4. *Transition* initiates the final preparations for making the system production-ready.

2.2 Workflows

The activities involved in building a system tend to be cohesive in terms of their interactions with other activities as well as the deliverables that are produced. These groupings are called workflows. The Unified Process refers to workflows as disciplines, but the two terms essentially mean the same thing. There are two types of workflows, Core Workflows and Supporting Workflows. Any given workflow occurs across a number of phases. Table 1 - Methodology Workflows identifies the workflows.

Table 1 - Methodology Workflows

Core Workflows	Supporting Workflows
1. Requirements	6. Configuration & Change Management
2. Architecture	7. Project Management
3. Realization	8. Environment
4. Verification	
5. Deployment	

The objectives and purpose of each workflow are detailed below.

2.2.1 Requirements

The Requirements Workflow assesses the needs and problems that are to be solved by the system and aim development toward the right system. This is achieved by describing the system's functional and nonfunctional requirements well enough so that an agreement can be reached between the Requirements Owner and the project team on what the system should and should not do. The results of the Requirements Workflow also help the Project Manager plan the iterations and customer releases.

2.2.2 Architecture

The Architecture Workflow lays the foundation for the system that will be built, ensuring it will meet its functional and nonfunctional requirements. The Architect will define the style and high-level outline of the architecture and decompose it into cohesive parts. The Architect with help from the project team will also identify and resolve the key technical risks through the creation of an evolutionary architectural prototype.

2.2.3 Realization

The Realization Workflow performs analysis and design, followed by coding and unit, and integration testing the implementation of the system. This results in UML models, source code, and executables.

2.2.4 Verification

The Verification Workflow plans the system and user-accepting tests required for each iteration, and executes the various tests to ensure that the system satisfies the requirements. This results in a system ready for deployment.

2.2.5 Deployment

The Deployment Workflow prepares for final delivery to the end users, including training, manuals, and hardware migration or data conversion.

2.2.6 Configuration & Change Management

The Configuration & Change Management Workflow defines and manages the configuration management and change control processes throughout the project. Change control provides the necessary tracking of changes between baselines in an iterative and incremental approach.

2.2.7 Project Management

The Project Management Workflow plans the project and iterations, making estimates, tracking costs, allocating resources, and managing risks.

2.2.8 Environment

The Environment Workflow plans, sets-up and maintains the various environments (i.e. development, test, staging, production, etc.) needed for the project. It also involves administration of any tools (i.e. tools for modeling, requirements management, code development, testing, etc.) to aid in developing and operating the system.

2.3 Iterations

The iterative approach breaks down a large project into a succession of mini-projects. Each mini-project is an iteration that results in an increment and allows you to address some requirements and some risks. You design a little, implement a little, validate it, and then take on more requirements, design some more, build some more, validate, and so on until you are finished.

Within each phase, the phase activities are carried out through a series of iterations. Each iteration implements a set of related use cases or mitigates some risks. During an iteration the project team proceeds through a series of workflows.

Figure 1 - Comparison of Sequential and Iterative Processes compares the iterative approach with the sequential approach.
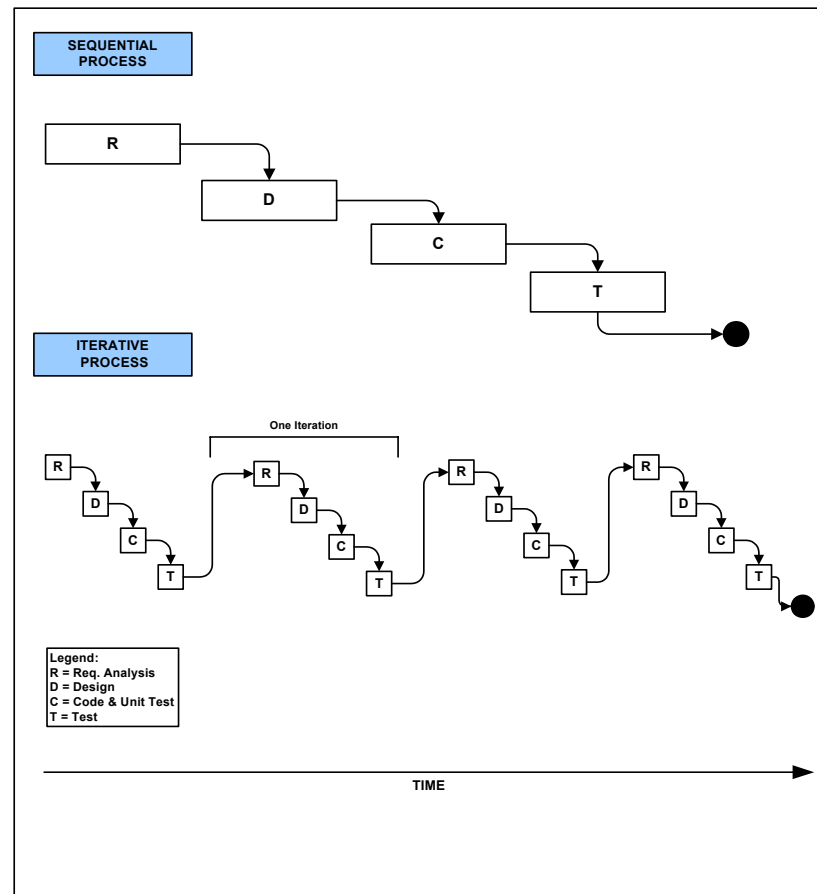
Figure 1 - Comparison of Sequential and Iterative Processes

Each iteration follows a pattern similar to the traditional waterfall or sequential approach. However, the emphasis on the various activities changes. In the Inception Phase, the focus is on understanding the overall requirements and determining the scope. The Elaboration Phase's focus is primarily on fleshing out requirements, but some software design and development is aimed at prototyping the architecture, mitigating the risks by trying solutions, and learning how to use certain tools and techniques. In the Construction Phase, the focus is mostly on design and development. In this phase the evolutionary architectural prototype from Inception and Elaboration evolves into the first operational product. In the Transition Phase, the focus is on ensuring that the system meets the users objectives; bugs are fixed, users trained and the final product delivered.

The benefits of an iterative approach compared with the traditional waterfall or sequential process are:

- Risks are mitigated earlier.

- Missing or unclear requirements may be caught sooner.

- Quality considerations are an integral and early part of the development.

- Change is more manageable.

- There is potential for a higher level of reuse.

The number of iterations on a project can vary depending on size, resource effort and schedule. Table 2 - Guidelines for Phases and Iterations shows the typical number of iterations, percentage of effort expended, and schedule consumed during each phase.

Table 2 - Guidelines for Phases and Iterations

	Inception	Elaboration	Construction	Transition
Low Number of Iterations	1	1	1	1
Medium Number of Iterations	1	2	3	1
High Number of Iterations	1	2	6	2
Effort	5%	20%	65%	10%
Schedule	10%	30%	50%	10%

2.4 Bringing It All Together

Figure 2 - Bringing It All Together brings together the phases, workflows and iterations and also depicts how the amount of effort in each workflow varies in each phase.
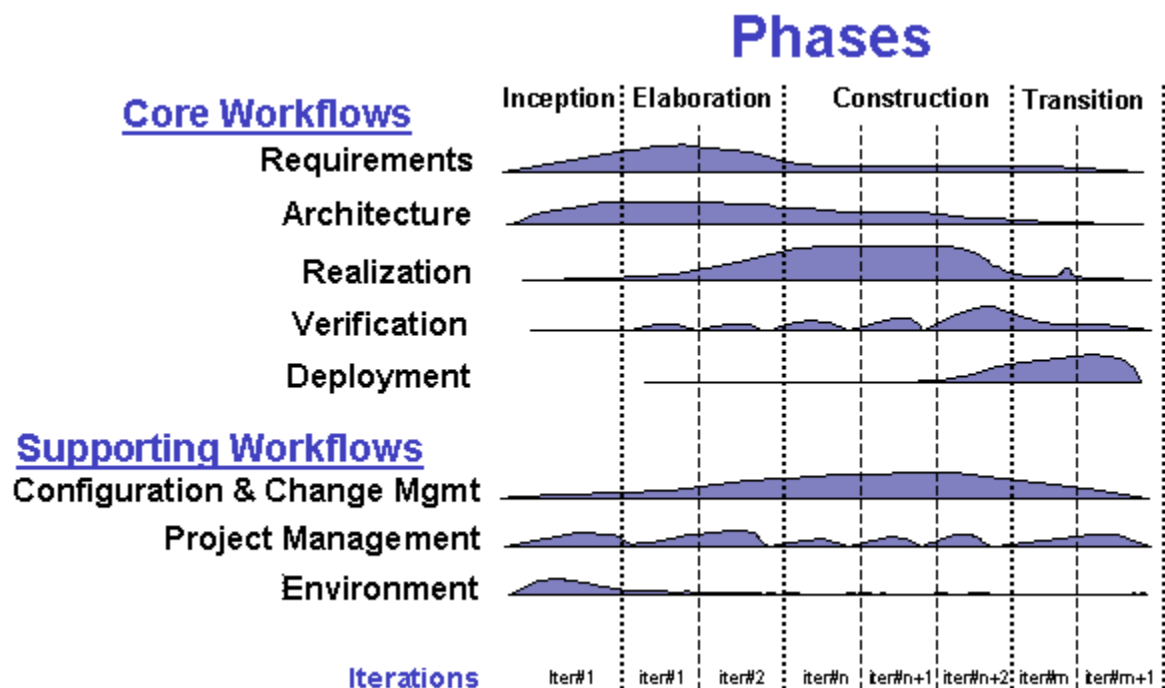
Figure 2 - Bringing It All Together

2.5 Tailoring to Fit the Project

Not all projects are alike; therefore, they cannot all follow the same Methodology. After a project is identified a representative from the PMO meets with the Project Manager to understand and document the profile of the project using the Project Profile Template. This allows tailoring of the Methodology requirements based on project size, scope, schedule and resources, etc.

3. Inception

3.1 Definition

Inception is the first phase. It defines the scope of the project, its risks, and estimates its major milestones. Understanding the scope requires a certain amount of exploration and documentation of the system's requirements, but only enough to prepare for the next phase. Inception formalizes the business vision, estimates what it will take to get there, and outlines how to know when and if success has been achieved.

3.2 Goals

The principle objective of the Inception Phase is to establish the business case for proceeding with the project. To establish this, we must define the scope of the proposed system, sketch out an architecture, identify the risks critical to the projects success, and outline a plan to mitigate those risks.

3.3 Exit Criteria

The following table identifies all the workflows and the activities and deliverables that are executed in the Inception Phase.

Table 3 - Inception Phase Activities and Deliverables

Workflow	Activity	Role
Requirements	✓ Vision Document completed, reviewed, baselined and signed off	Requirements Analyst with input from Requirements Owner and Business Owner
	✓ Requirements Document initiated, reviewed and baselined	Requirements Analyst with help from the Architect and input from Requirements Owner
Architecture	✓ System Architecture Document initiated, reviewed and baselined	Architect
Realization	✓ Optional throw-away well-scoped proof-of-concept prototype created	Developer with help from Architect
Verification	--------------None Prescribed-------------	-------------NA-----------
Deployment	————None Prescribed————	————NA————
Configuration & Change Management	✓ Configuration and Change Management Plan initiated, reviewed and baselined	CCM Analyst with help from Project Manager
Project Management	✓ Project Plan created, reviewed and baselined	Project Manager with help from Architect
	✓ Iteration Plan created, reviewed and baselined for iterations in Elaboration	Project Manager with input from Developers
	✓ Risk List is created and risks are managed and tracked	Project Manager with input from Architect
	✓ Project status is coordinated, tracked and documented	Project Manager

Environment	✓ Elaboration environments are planned and implemented	System Analyst with help from Architect
	✓ Tool selection is documented	Architect with input from project team and help from Project Manager

3.3.1 Requirements Workflow

During the Inception Phase, the focus is on the identification of the major business and information needs of the proposed system and the definition of the scope of the project in the Vision Document. Additionally, functional and nonfunctional requirements and an initial Domain Model are captured in the Requirements Document. User feedback is evaluated and assessed at the end of the Inception Phase.

The Vision Document is completed, reviewed, baselined and signed off

This document defines the scope of the project. It includes the business context, system context, stakeholders and constraints. It also includes prioritized Systemic Qualities and functional requirements that the proposed system must meet.

Requirements Document initiated, reviewed and baselined

This document contains the primary detailed requirements of the proposed system. It includes constraints, Service Level Requirements (SLRs), Domain Model (DM) and business rules, both business and operational functional requirements, and a glossary. By the end of the Inception Phase, the following RD sections must be completed:

- ❑ Actor list and description 80%
- ❑ Use cases list and brief description 90%
- ❑ Use case detail description 10-20%
- ❑ Use case model (optional) 100%
- ❑ Key entities list and description 100%
- ❑ DM and business rules 80%
- ❑ Constraints 80%
- ❑ Glossary 80%
- ❑ SLRs 80%

Please note that the percentages listed above are an approximation of the percentage complete.

3.3.2 Architecture Workflow

During the Inception Phase, the architecture should be resolved to a level of detail sufficient to establish a reasonable degree of confidence in the technical scope. The architecture should also be detailed enough to define the first Elaboration Iteration Plan.

System Architecture Document initiated, reviewed and baselined

This document contains the primary architectural description of the proposed system. This includes the structure and behavior of the system that meets the architectural requirements as specified in the Requirements Document. It also includes the reasoning and motivation for decisions chosen. When applicable, this document also describes how the system has been designed to handle evolution over time.

By the end of the Inception Phase, the following SAD sections must be completed:

- ❑ Key risk areas mapped to identified use cases 100%
- ❑ Enterprise perspective 60%

❑ High-Level View of the architecture 80 – 90%

❑ Selection and reasoning of mechanisms and platforms in the Upper Platform Layer 80%

❑ SLR solution alternatives 60-80%

3.3.3 Realization Workflow

During the Inception Phase, proof-of-concept prototypes may be developed to find and demonstrate a possible solution to a business or technical problem.

Optional throwaway well-scoped proof-of-concept prototype created

This prototype both implements and illustrates an idea of the new system with focus on its use, user interfaces, integration with other systems, and/or some interesting new algorithms. This activity is optional and is rarely executed. It may also be undertaken in the Elaboration Phase as well, but, if done then, should be minimal.

3.3.4 Configuration & Change Management Workflow

During the Inception Phase, the Configuration & Change Management Plan (CCMP) is developed and change management procedures and standards are established including: change control, version control, builds, baselines, releases and a Change Control Board (CCB).

Configuration & Change Management Plan initiated, reviewed and baselined

This document describes the configuration and change management activities to be performed on the project. For the most part, the CCMP references the EOHHS CCM standards published by the PMO for policies, procedures and standards to be used by the project team to manage and control the project's deliverables. The CCMP also indicates any deviations from the EOHHS CCM standards and any project specific information such as release names, etc.

3.3.5 Project Management Workflow

During the Inception Phase, the Project Management Workflow establishes the overall project organization, understands risks, eliminates unknowns and develops the initial project estimates in a Project Plan.

Project Plan created, reviewed and baselined

The Project Plan is created using the Project Plan template. It defines the following:

❑ Project Methodology

❑ Project Schedule

❑ Roles and Responsibilities

❑ Deliverables

❑ Stakeholders

❑ Review Process

❑ Checklists

❑ Issue Resolution

❑ Document Change Process

❑ Acceptance Plan

The Project Schedule that is part of the Project Plan encompasses all the iterations in all of the phases of the project. It is driven by use cases and tasked by workflows. It identifies the milestones, deliverables and tasks. For each task, the following is specified:

- ❑ Owner

- ❑ Start Date

- ❑ End Date

- ❑ Duration

- ❑ Dependencies

The Project Schedule template identifies all activities and deliverables defined in this user guide and can be tailored by the Project Profile to fit the size, scope and schedule of each project.

In the Inception Phase, the project budget must be determined in addition to the Project Schedule in terms of major milestones, list of iterations & their goals, list of releases, demos, etc.

All information described in the Project Plan and Project Schedule is based on data available to-date. Prior to the start of each phase the Project Plan including the Project Schedule is updated with known information, refined resources and time estimates, reviewed, and re-baselined.

Iteration Plan created, reviewed and baselined for iterations in Elaboration

An Iteration Plan encompasses activities and deliverables for one iteration. For each iteration in Elaboration, an Iteration Plan needs to be created. The Iteration Plan is tasked by workflow and shows:

- ❑ Use cases and scenarios to be developed in the iteration

- ❑ Timelines and intermediate milestones for the iteration, such as when testing starts, beta version, demos etc.

- ❑ Resources needed for the iteration

- ❑ Evaluation criteria such as functionality, quality goals, etc.

Iterations should be fairly short in duration. Goals and work breakdown of one iteration are defined during the previous iteration(s). Selection of work done in initial iterations is driven by risk. As risks become tamed, functionality considerations predominate. Since the Project Schedule reflects iterations, it is revisited after each iteration.

Risk List is updated and risks are managed and tracked

During all phases of the project, the Project Manager and the appropriate project team members meet regularly to identify, assess and manage any new risks associated with the project. All project risks are documented and tracked in the Risk List and reported in the project's weekly Status Report.

Project status is coordinated, tracked and documented

The Project Manager is responsible for preparing the weekly project Status Report. Information received at the project status meeting and throughout the week is documented in the project Status Report. In addition the Status Report documents and tracks the project's risks and logs any date changes to deliverables and milestones along with the reason for those changes. Copies of the weekly Status Reports are distributed to all project team members and others as requested.

The Project Manager, at his/her discretion, may request that the project team members provide individual Status Reports.

3.3.6 Environment Workflow

During the Inception Phase, environment planning begins.

Elaboration environments planned and implemented

This activity involves establishing the hardware and software environments needed for the Elaboration Phase specifically for the evolutionary architectural prototype. The environment requirements are documented in the 2 Configurations View sections of the System Architecture Document (SAD).

The following environments are planned and implemented:

- ❑ Development
- ❑ Integration testing
- ❑ System testing

Tool selection is documented

Tools may be selected that fit the particular needs of the project. For example, tools for modeling, code development (editors, compilers, debuggers), configuration management, testing, planning and tracking, and document preparation. Tool selection is done in accordance with the ITA and is documented in the Enterprise Perspective section of the System Architecture Document.

4. Elaboration

4.1 Definition

Elaboration brings the architecture to the point where the construction of the system can proceed efficiently. With risk management being a primary focus of Elaboration, the now well-defined architecture can be documented through evolutionary architectural prototyping. Further requirements development is also a key activity and the use cases are now expanded to include the detail. Elaboration is truly complete only when there are no predictable significant risks or unknowns remaining.

4.2 Goals

Elaboration defines the architecture, which serves as the foundation for subsequent development. It also creates an architectural baseline that includes the architecturally significant functionality of the system and features important to the customer. Other key goals of Elaboration include addressing significant risks, and further detailing the functional and nonfunctional requirements of the system.

4.3 Exit Criteria

The following table identifies all the workflows and the activities and deliverables that are executed in the Elaboration Phase.

Table 4 - Elaboration Phase Activities and Deliverables

Workflow	Activity	Role
Requirements	✓ Requirements Document augmented, reviewed, baselined and signed off	Requirements Analyst with help from the Architect and input from Requirements Owner
Architecture	✓ System Architecture Document augmented, reviewed, baselined and signed off	Architect
Realization	✓ Evolutionary Architectural Prototype designed, built, and unit and integration tested	Developer with supervision from Architect
Verification	✓ Evolutionary Architectural Prototype system tested	Test Analyst with supervision from Architect
	✓ Test Plans initiated, reviewed and baselined	Developer and Test Analyst with input from Architect and Requirements Owner
	✓ Integration, System and UAT Test Cases are identified	Developer and Test Analyst with input from Architect and Requirements Owner
Deployment	✓ Deployment Plan initiated, reviewed and baselined	Project Manager with input from project team
Configuration & Change Management	✓ Configuration and Change Management Plan augmented, reviewed, baselined and signed off	CCM Analyst with help from Project Manager
Project Management	✓ Project Plan updated, reviewed and baselined	Project Manager with help from Architect
	✓ Iteration Plan created, reviewed and baselined for Construction iterations	Project Manager with input from Developers
	✓ Risk List is updated and risks are managed and tracked	Project Manager with input from Architect
	✓ Project status is coordinated, tracked and documented	Project Manager
Environment	✓ Construction environments are planned and implemented	System Analyst with help from Architect
	✓ Tools are installed and configured	System Analyst with help from Architect

4.3.1 Requirements Workflow

During the Elaboration Phase, the principle objective of the Requirements Workflow is to capture the remaining requirements. User feedback is evaluated and assessed at the end of the Elaboration Phase.

Requirements Document augmented, reviewed, baselined and signed off

This activity involves the further development and refining of requirements and results in the updating of the Requirements Document that was started in the Inception Phase. By the end of the Elaboration Phase, the RD should be completed including the following sections:

- ❑ Actor list and description 100%
- ❑ Use case detail description 80%
- ❑ DM and business rules 100%
- ❑ Constraints 100%
- ❑ Glossary 100%
- ❑ SLRs 100%

Additional use cases, if any, may be added to the RD.

4.3.2 Architecture Workflow

During the Elaboration Phase, the principle objective of the Architecture Workflow is to establish a sound architectural foundation for the Construction Phase.

System Architecture Document augmented, reviewed, baselined and signed off

This activity involves the updating of the System Architecture Document (SAD) that was started in the Inception Phase. All sections on which work was begun during Inception are finished. In addition, several major activities are begun during the Elaboration Phase. By the end of the Elaboration Phase, the SAD should be completed including the following sections:

- ❑ Enterprise perspective 100%
- ❑ High-Level View of the architecture 100%
- ❑ Selection and reasoning of mechanisms and platforms in the Upper Platform Layer 100%
- ❑ SLR solution alternatives 100%
- ❑ All views detailed 100%
    - o Decomposition & dependencies
    - o Configurations across layers
    - o How system functionality is met
    - o How risks have been mitigated
    - o How SQs have been addressed
    - o Guidelines for designers

o

4.3.3 Realization Workflow

During the Elaboration Phase, the focus of the Realization Workflow is to develop and test the evolutionary Architectural Prototype that will be extended in the Construction Phase.

Evolutionary Architectural Prototype designed, built, and unit and integration tested

This activity involves the designing, building and unit, and integration testing of the Evolutionary Architectural Prototype. This prototype consists of the implementation of actual end user functionality in the form of working use cases. The main reason for implementing the use cases is to mitigate the targeted risks. The prototype incrementally evolves into the complete system over the course of the project.

The main activities and deliverables include:

- ❑ Analysis and design models

- ❑ Code with code comments

- ❑ Unit and integration test harness

- ❑ Unit testing conducted

- ❑ Integration testing conducted after each iteration

- ❑ Unit and integration test results reported

The Architect, at his/her discretion, may decide whether Developers should maintain their analysis and/or design models. The Architect also decides the code and design review approach.

4.3.4 Verification Workflow

The objectives of the Test Workflow during the Elaboration Phase are to develop the Test Plans and Test Cases for Integration, System and UAT testing, to test the evolutionary architectural prototype and provide the test results.

- *Unit Testing* exercises a relatively small executable. In object-oriented programming, an instance of a class is the smallest executable unit. A test unit may be a class or several related classes.

- *Integration Testing* is performed to verify that the components integrated into a build operate together properly. The result of integration test is a complete system or subsystem of software and hardware units.

- *System Testing* is exercised on a completely integrated application. Tests are conducted on the capabilities and characteristics that are only present in the entire system.

- *User Acceptance Testing (UAT)* is conducted to determine whether or not a system satisfies the acceptance criteria of the Requirements Owner.

Service Level Requirements, such as Performance, are reviewed and included for validation in Integration, System and UAT Test Plans.

Regression testing is selective retesting of the system or component to verify that modifications have not caused unintended effects and that previously-working functionality has not been adversely affected by recent changes. Regression testing is required as a normal function of Unit, Integration, System and UAT testing.

Test Plans initiated, reviewed and baselined

The following Test Plans are initiated for:

- Integration testing

- System testing (functionality and SLRs)

- User Acceptance testing (UAT)

The Test Plans for Integration, System and User Acceptance Testing (UAT) define the scope, objective, assumptions and constraints, strategy, entry/exit criteria, pass/fail criteria, test configuration, regression test strategy, and the resources needed to adequately test the system. The Test Plans also describe the problem tracking and control process that is used throughout testing. Developer(s) are responsible for Integration Test planning. The Test Analyst with the Architect's supervision is responsible for System Test planning, and the Requirements Owner and/or the Test Analyst are responsible for UAT planning.

Evolutionary Architectural Prototype system tested

System testing of the Evolutionary Architectural Prototype is conducted after each iteration with account for regression testing. System testing addresses both functional and Service Level Requirements. System test results should be reported.

Integration, System and UAT Test Cases are identified

A test case defines a specific way of testing the system. Test cases are derived from use cases. A test case includes a set of inputs, execution conditions, and expected results developed for a particular objective. In addition it defines pre and post test states of the environment. The test cases are started in this phase and continue into the Construction Phase. There are 4 types of test cases:

- *An Integration Test Case* validates that the software and hardware units that make up a component are operable. The Developer is responsible for identifying and developing the Integration Test Cases and for conducting integration testing.

- *A System Test Case* falls in one of 2 categories:

    o *Test Cases for Use Cases* specify how to test a Use Case. This test case verifies that the results of the interactions between the actors and the system are as expected, that the pre- and post-conditions specified by the Use Case are satisfied, and that the sequence of actions specified by the Use Case is followed. The Test Analyst is responsible for identifying and developing System Test Cases and for executing system testing.

    o *Test Cases for SLRs*, such as Performance, are also identified and developed to ensure that they are implemented as stated in the Vision Document, Requirements Document and System Architecture Document. The Test Analyst with help from the Project Manager, Architect, Developer and System Analyst is responsible for identifying and developing Test Cases for the SLRs and for executing SLR testing as part of system testing.

- *A User Acceptance Test (UAT) Case* specifies how to test a specific business process scenario of a Use Case. The Test Analyst and/or Requirements Owner are responsible identifying for developing UAT Cases and for conducting the UAT.

4.3.5 Deployment Workflow

The objective of the Deployment Workflow during the Elaboration Phase is to uncover risk and plan out strategies around hardware migration and data conversion in preparation for deployment.

Deployment Plan initiated, reviewed and baselined

The Deployment Plan describes the strategy to be used for the conversion of the appropriate data, the deployment of the software and hardware components, the infrastructure requirements of the deployment process, the plan for rollout, rollout contingency plans, training strategy and needed resources.

Sections of the Deployment Plan pertaining to the migration of hardware or data conversion should be started during the Elaboration Phase, since they often pose risks to the project. They may involve considerable lead-time in terms of notifying, interacting, and formulating appropriate strategies in tandem with the appropriate organizations.

4.3.6 Configuration & Change Management Workflow

During the Elaboration Phase, the Configuration & Change Management Workflow continues to manage the change control procedure such as compiling the code into builds and conducting issue and bug reviews, etc.

Configuration and Change Management Plan augmented, reviewed, baselined and signed off

The CCMP is augmented with additional information not known during Inception. The CCMP references the EOHHS CCM standards and includes the information needed to manage and control the project's deliverables and defines the control of document baselines. It defines the use and access to the software control tool and the control of software baselines, builds, and releases.

4.3.7 Project Management Workflow

During the Elaboration Phase, the principle objectives of the Project Management Workflow are to continue to monitor the remaining critical risks and to identify any significant new risks to the point that we can estimate their impact on the project, to further detail the Project Plan and to plan the Construction iterations.

Project Plan updated, reviewed and baselined

At the end of each iteration the Project Plan is updated with known actuals and refined resource and time estimates.

Iteration Plan created, reviewed and baselined for Construction iterations

Plans for the iterations of the Construction Phase are developed, including the goals of each iteration and the use cases that will be developed in each iteration. These tasks should outline the sequence of builds and the tasks for integration of the hardware and software components.

Risk List is updated and risks are managed and tracked

During all phases of the project, the Project Manager with the assistance of the appropriate project team members meet regularly to identify, assess and manage any new risks associated with the project. All project risks are documented and tracked in the Risk List and reported in the project's weekly Status Report.

Project status is coordinated, tracked and documented

Project status reporting continues as in Inception.

4.3.8 Environment Workflow

In the Elaboration Phase, the Environment Workflow focuses on planning and implementing the environments for the Construction Phase. Tools are installed and configured, and users are trained to use them.

Construction environments are planned and implemented

This activity involves establishing the hardware and software environments needed for the Construction Phase in order to build the system. The environment requirements are documented in the 2 Configurations View sections of the System Architecture Document (SAD).

The following environments are planned implemented:

- ❏ Development
- ❏ Integration testing
- ❏ System testing (functionality and SLRs)

All existing environments are managed. Managing the environments includes maintaining the hardware and software, performing system administration, backups, and upgrades.

Tools are installed and configured

The selected tool(s) must be acquired, installed and configured during the Elaboration Phase. In addition, the users must be trained to use them.

5. Construction

5.1 Definition

The system is built during Construction. This does not imply that development occurs only during Construction (it occurs before and after as well), but the bulk of development occurs during Construction. The Construction Phase is complete when the system can be given to the Requirements Owner for User Acceptance Testing.

5.2 Goals

The approach to Construction is incremental and iterative. During the Construction Phase the use case detailed descriptions, analysis, design, coding and testing are finished while risks are continuously monitored. This approach results in a system that is implemented as a succession of small and manageable VG and ensures that the system is ready to be released to UAT and production.

5.3 Exit Criteria

The following table identifies all the workflows and the activities and deliverables that are executed in the Construction Phase.

Table 5 - Construction Phase Activities and Deliverables

Workflow	Activity	Role
Requirements	✓ Requirements Document updated, reviewed, baselined and signed off	Requirements Analyst with help from the Architect and input from Requirements Owner
Architecture	✓ System Architecture Document updated, reviewed, baselined and signed off	Architect

Realization	✓ Functional system designed, built and unit, and integration tested	Developer with supervision from Architect
Verification	✓ System test is performed	Test Analyst with supervision from Architect
	✓ Test Plans augmented, reviewed, baselined and signed off	Developer and Test Analyst with input from Architect and Requirements Owner
	✓ Integration, System and UAT Test Cases are augmented, reviewed, baselined and signed off	Developer and Test Analyst with input from Architect and Requirements Owner
Deployment	✓ Deployment Plan augmented, reviewed, baselined and signed off	Project Manager with input from project team
	✓ User and Operational Manuals and training material initiated, reviewed and baselined	Technical Writer and Training Department with help from the project team
	✓ Service Level Agreement determined, initiated, reviewed and baselined	Project Manager with input from project team
Configuration & Change Management	✓ Configuration and Change Management Plan updated, reviewed, baselined and signed off	CCM Analyst with help from Project Manager
Project Management	✓ Project Plan updated, reviewed and baselined	Project Manager with help from Architect
	✓ Iteration Plan created, reviewed and baselined for Transition iterations	Project Manager with input from Developers
	✓ Risk List is updated and risks are managed and tracked	Project Manager with input from Architect
	✓ Project status is coordinated, tracked and documented	Project Manager
Environment	✓ Transition environments are planned and implemented	System Analyst with help from Architect

5.3.1 Requirements Workflow

The main goal of the Requirements Workflow in the Construction Phase to complete the final 20% of the use case details, resolve requirements ambiguities, and update the RD as change requests are approved. User feedback is evaluated and assessed at the end of the Construction Phase.

Requirements Document updated, reviewed, baselined and signed off

The use case detailed descriptions should be 100% complete. If any change requests are approved, they are added to the RD.

5.3.2 Architecture Workflow

During the Construction Phase, updates to the System Architecture Document are managed through the change control process.

System Architecture Document updated, reviewed, baselined and signed off

Any final changes are made to the SAD. It is updated to reflect the results of the SLR testing.

The Architect communicates and oversees the architecture integrity as the Developers build the system.

5.3.3 Realization Workflow

The Realization Workflow is the key focus during the Construction Phase to develop the system's functionality. Modules are designed, built, and tested according to the Iteration Plan.

Functional system designed, built and unit, and integration tested

This activity involves extending the Evolutionary Architectural Prototype to build the final system. Development adheres to the Iteration Plan and to the architecture.

The main activities and deliverables include:

- ❑ Design and design reviews
- ❑ Code with code comments and code reviews
- ❑ Unit and integration test harness
- ❑ Unit testing conducted
- ❑ Integration testing conducted after each iteration
- ❑ Unit and integration test results reported
- ❑ Bugs are fixed as needed
- ❑ Installation test harness

As in the Elaboration Phase, the Architect, at his/her discretion, may decide whether Developers should maintain their analysis and/or design models. The Architect also decides the code and design review approach.

5.3.4 Verification Workflow

During the Construction Phase, system testing is executed to ensure the system is ready for User Acceptance Testing (UAT) in Transition. The test cases are finalized and signed off.

System test is performed

System Test is conducted with account for regression testing to ensure that all the specified use cases and Service Level Requirements are verified and to establish the system's readiness for release to UAT and production. System testing is done at the end of every iteration during the Construction Phase. Test results are reported. Any issues and bugs discovered as a result of testing are fixed and retested.

Test Plans augmented, reviewed, baselined and signed off

The Integration, System and UAT Test Plans are completed, reviewed and signed off.

Integration, System and UAT Test Cases are augmented, reviewed, baselined and signed off

The Integration, System and User Acceptance Test (UAT) Test Cases identified in the Elaboration Phase are developed and reviewed for completeness.

5.3.5 Deployment Workflow

During the Construction Phase, the Deployment Plan is completed for the Transition Phase. User and Operational Manuals and training material are created. If needed, SLAs are also created.

Deployment Plan augmented, reviewed, baselined and signed off

The infrastructure requirements of the deployment process, strategy for rollout, rollout contingency plans, training strategy and needed resources are added to the Deployment Plan. Updates to the migration of hardware or to the data conversion requirements are also completed.

User and Operational Manuals and training material initiated, reviewed and baselined.

Initial User and Operational Manuals are created to assist the users and operators with the systems functionality. Training materials such as slides, online tutorials or computer-based training (CBT) are created.

All user and operational documentation is verified to ensure it meets its intended use. These tests are conducted prior to release.

Service Level Agreement determined, initiated, reviewed and baselined

If a need for a SLA is established, the document is initiated. The SLA dictates the hardware or platform where the software resides, severity levels, response times, escalation times, hours of support, resolution procedures and sign-off by all affected parties.

5.3.6 Configuration & Change Management Workflow

During the Construction Phase, the Configuration & Change Management Workflow continues to manage the change control procedures such as issues and bug reviews, etc.

Configuration and Change Management Plan updated, reviewed, baselined and signed off

Final changes may be made to the CCMP as per the change control procedures.

5.3.7 Project Management Workflow

During Construction, Project Management involves a higher emphasis on coordination and control, due to the greater number of resources and tasks needed for each iteration.

Project Plan updated, reviewed and baselined

During Construction, at the end of each iteration the Project Plan is updated with known actuals and refined resource and time estimates.

- ❑ Assess progress to plan
- ❑ Possibly re-order use cases
- ❑ Deal with added functionality
- ❑ Identify and plan rework
- ❑ Make resource adjustments as needed

Iteration Plan created, reviewed and baselined for Transition iterations

Plans for the remaining Transition iteration(s) are developed, including the goals for each iteration.

Risk List is updated and risks are managed and tracked

By the end of Construction, all risks should be mitigated. During all phases of the project, the Project Manager with the assistance of the appropriate project team members meet regularly to identify, assess and manage any new risks associated with the project. All project risks are documented and tracked in the Risk List and reported in the project's weekly Status Report.

Project status is coordinated, tracked and documented

Project status reporting continues as in the previous phases.

5.3.8 Environment Workflow

In the Construction Phase, the Environment Workflow focuses on planning and implementing the environments for the Transition Phase.

Transition environments are planned and implemented

This activity involves establishing the hardware and software environments needed for Transition in order to deploy the system. The environment requirements are documented in the 2 Configurations View sections of the System Architecture Document (SAD).

The following environments are planned implemented:

- ❑ UAT
- ❑ Production Staging
- ❑ Production

All existing environments are managed. Managing the environments includes maintaining the hardware and software, performing system administration, backups, and upgrades.

6. Transition

6.1 Definition

Transition initiates the final preparations for a full release. User Acceptance Testing is performed during the Transition Phase. The focus of the team changes, as the end users are more intimately involved with system progression. Those users must be supported; bugs must be tracked and fixed, minor enhancements made and all the support material (manuals, training, installation, etc.) must be completed. All project documentation such as the RD and the SAD must be up-to-date. Transition ends with a formal release.

6.2 Goals

Transition allows the end users to test the system, the manuals and other documentation that are prepared for the product release.

6.3 Exit Criteria

The following table identifies all the workflows and the activities and deliverables that are executed in the Transition Phase.

Table 6 - Transition Phase Activities and Deliverables

Workflow	Activity	Role
Requirements	--------------None Prescribed-------------	------------NA------------
Architecture	————None Prescribed————	————NA————
Realization	✓ Bug fixes and minor enhancements are added as needed	Developer with supervision from Architect
Verification	✓ UAT is performed	Requirements Owner with help from Test Analyst
Deployment	✓ Deployment Plan is executed	Project Manager with input from project team
	✓ User and Operational Manuals and training material augmented, reviewed, baselined and signed off	Technical Writer and Training Department with help from the project team
	✓ Service Level Agreement augmented, reviewed, baselined and signed off	Project Manager with input from project team
Configuration & Change Management	✓ Change requests are assessed	Project Manager with input from project team
Project Management	✓ Project Plan updated, reviewed and baselined	Project Manager with help from Architect
	✓ Risk List is updated and risks are managed and tracked	Project Manager with input from Architect
	✓ Project status is coordinated, tracked and documented	Project Manager
Environment	✓ Environments are managed	System Analyst with help from Architect

6.3.1 Realization Workflow

During the Transition Phase, bugs and enhancements are fixed as required.

Bug fixes and minor enhancements are added as needed

In this phase, the Developer(s) fixes minor bugs and makes the necessary enhancements as defined by the change management procedure. The system is retested as needed.

6.3.2 Verification Workflow

User Acceptance Testing is performed during the Transition Phase. A Beta/Pilot test may be conducted if necessary. User Acceptance Testing (UAT) is conducted to determine whether or not a system satisfies its acceptance criteria. UAT consists of test scenarios that incorporate all of the Use Cases needed to validate the requirements and to meet success criteria established for the project. UAT may also include a regression testing where necessary.

UAT is performed

UAT testing is performed at each iteration in Transition to verify that the requirements have been satisfied, all of the major risks have been addressed and the system's current functionality has not been affected by the new and/or changed functionality and that it interfaces properly with existing systems. Any issues or bugs discovered as a result of test are fixed and re-tested. The test results re reported.

6.3.3 Deployment Workflow

During the Transition Phase, the system is deployed according to the Deployment Plan.

Deployment Plan is executed

This activity moves the software and hardware components to production. This activity is monitored to ensure that there are no adverse effects to the current production environment or in the ability of the user to perform his/her daily tasks. The Project Manager is responsible for monitoring the deployment activity.

User and Operational Manuals and training material augmented, reviewed, baselined and signed off

User and operational documentation is finalized and distributed. Training is conducted. Training consists of formal classroom training, online tutorials or computer-based training (CBT).

Service Level Agreement augmented, reviewed, baselined and signed off

Service Level Agreements are finalized at least one week prior to the release of the system into a production environment.

6.3.4 Configuration & Change Management Workflow

During the Transition Phase, change requests are identified as "fix now" or "features for future releases."

Change requests are assessed

Assess change requests to be implemented in the current or future project cycle and per the change control procedures.

6.3.5 Project Management Workflow

The objectives of the Project Management Workflow during Transition are to manage team activities according to the Project Plan to ensure everything is ready to meet the scheduled deployment date. Upon project shutdown, the lessons learned are documented as well as the next release goals. The stakeholders signoff on the final release.

Project Plan updated, reviewed and baselined

The Project Plan is updated to reflect changes in resources, iterations, cost, etc.

Risk List is updated and risks are managed and tracked

During all phases of the project, the Project Manager with the assistance of the appropriate project team members meet regularly to identify, assess and manage any new risks associated with the project. All project risks are documented and tracked in the Risk List and reported in the project's weekly Status Report.

Project status is coordinated, tracked and documented

Project status reporting continues as in the previous phases.

6.3.6 Environment Workflow

All required environments are managed. Production environments are managed as determined by the SLAs.

Environments are managed

The required environments are managed such as:

❑ Development

❑ Integration testing

❑ System testing

❑ UAT

❑ Production Staging

❑ Production

Managing the environments includes maintaining the hardware and software, performing system administration, backups, and upgrades

7. Glossary

The following table defines terms and acronyms used in this user guide.

Term	Definition
CCB	Change Control Board
CCMP	Configuration & Change Management Plan
DM	Domain Model
PMO	Project Management Office
RD	Requirements Document
SAD	System Architecture Document
SLA	Service Level Agreement
SLR	Service Level Requirement
SQ	Systemic Quality
VG	Virtual Gateway
UAT	User Acceptance Test
UP	Unified Process
VD	Vision Document

8. Tools & Job Aids

The following table references templates and guides to assist the project team apply the Methodology. Please note that the documents formatted in italic are not currently available as part of the latest EOHHS Methodology release.

Workflow	Title	Location
Requirements	Vision Document Template	
	Requirements Document Template	
	Use Case Template	
	Use Case Guidelines	
	Requirements Tutorial	
Architecture	System Architecture Document Template	
	Architecture Tutorial	
Realization	*Design & Code Review Process*	
Verification	*Test Plan Template*	
	Test Case Template	
Deployment	*Deployment Plan Template*	
	User and Operational Manuals Template	
	SLA Template	
Configuration and Change Management	*Configuration & Change Management Plan Template*	
	Change Control Procedures	
Project Management	Risk List Template	
	Project Profile Template	
	Project Plan Template	
	Project Schedule Template	
	Iteration Plan Template	
	Status Report Template	
	Deliverables Checklist	